



### 三种GNSS数据下载方法对比分析

吴汤婷, 梁 乔, 陈本富, 卢立果

#### Comparative analysis of three GNSS data download methods

WU Tangting, LIANG Qiao, CHEN Benfu, and LU Ligu

引用本文:

吴汤婷, 梁乔, 陈本富, 等. 三种GNSS数据下载方法对比分析[J]. *全球定位系统*, 2022, 47(3): 99–103. DOI: [10.12265/j.gnss.2021120602](https://doi.org/10.12265/j.gnss.2021120602)

WU Tangting, LIANG Qiao, CHEN Benfu, et al. Comparative analysis of three GNSS data download methods[J]. *Gnss World of China*, 2022, 47(3): 99–103. DOI: [10.12265/j.gnss.2021120602](https://doi.org/10.12265/j.gnss.2021120602)

在线阅读 View online: <https://doi.org/10.12265/j.gnss.2021120602>

### 您可能感兴趣的其他文章

#### Articles you may be interested in

#### 一种高效的GNSS数据自动化下载方法的实现

A High Efficient Method of Downloading GNSS Data Automaticly

*全球定位系统*. 2018, 43(3): 129–134

#### GNSS干扰源测向方法对比分析

Comparison and analysis of direction finding methods of GNSS interference sources

*全球定位系统*. 2019, 44(2): 59–62

#### 基于G-nut / Anubis的GNSS多系统数据质量分析研究

Data quality analysis of GNSS multi-system based on G-nut / Anubis

*全球定位系统*. 2020, 45(3): 28–32

#### GPS / GLONASS / BDS / Galileo系统载波相位观测值质量对比分析

Quality contrast and analysis of carrier phase observations in GPS / GLONASS / BDS / Galileo system

*全球定位系统*. 2019, 44(1): 16–22

#### 三种常用周跳探测与修复方法的性能分析

The performance analysis of three common methods of cycle slip detection and repair

*全球定位系统*. 2020, 45(5): 1–5

#### 北斗卫星导航系统动态定位精度测试与分析

Dynamic Positioning Accuracy Test and Analysis of BeiDou Satellite Navigation System

*全球定位系统*. 2018, 43(1): 43–48



关注微信公众号, 获得更多资讯信息

DOI: 10.12265/j.gnss.2021120602

# 三种 GNSS 数据下载方法对比分析

吴汤婷<sup>1,2</sup>, 梁乔<sup>2</sup>, 陈本富<sup>2</sup>, 卢立果<sup>3</sup>

(1. 东华理工大学 资源与环境经济研究中心, 南昌 330013;

2. 东华理工大学 测绘工程学院, 南昌 330013;

3. 桂林理工大学 广西空间信息与测绘重点实验室, 广西 桂林 541006)

**摘要:** 为综合评估单线程、多线程和多协程在全球卫星导航系统 (GNSS) 数据下载中的性能, 以武汉大学国际 GNSS 服务 (IGS) 数据中心为下载源, 从下载文件的时间跨度和量级大小两个方面对三种方法的下载效果进行对比分析。结果表明: 在中小型文件下载过程中多线程的优势大于单线程和多协程; 在大型文件下载中, 多协程的下载效果略优于多线程, 且都相较单线程性能更佳; 由于 GNSS 数据普遍是中小型文件, 所以多线程更适用于 GNSS 数据下载工作。

**关键词:** 全球卫星导航系统 (GNSS); 数据下载; 单线程; 多线程; 多协程

中图分类号: P228.4

文献标志码: A

文章编号: 1008-9268(2022)03-0099-05

## 0 引言

高精度全球卫星导航系统 (GNSS) 数据是定位、导航和授时 (PNT) 服务的前提和基础, 对最终服务精度的好坏起决定性作用。在 PNT 智能化服务大背景下<sup>[1]</sup>, 伴随高精度数据解算工作的深入, GNSS 数据的重要性愈加突显。一方面, GNSS 的服务范围不断拓宽, 已是部分传统技术手段的重要替代技术<sup>[2]</sup>, 相关数据解算成为业内人员的日常工作; 另一方面, GNSS 自身的发展和完善, 促进 GNSS 数据在种类和数量上迅速增长, 被广泛应用于电离层、对流层、参考框架、地球自转和地壳运动等研究<sup>[3-7]</sup>。高精度 GNSS 数据主要来源于国际 GNSS 服务 (IGS) 和连续运行参考站 (CORS), 供用户下载使用。

伴随用户对 GNSS 数据的需求急剧增长, 如何高效获取大批量 GNSS 数据成为高精度 PNT 服务的首要考虑问题。物理层面上, 可通过提升信道带宽来加快文件下载速率, 但这无疑为用户带来更高的成本; 技术层面上, 操控多台计算机共同执行相同下载任务能有效缩短耗时, 不足在于占用大量计算机资源。因此已有众多学者对如何高效下载 GNSS 数据进行研究, 主要集中在基于单线程下载的软件研发, 对多线

程和多协程下载方法涉及较少<sup>[8-11]</sup>。因此, 本文以自研软件为平台, 对单线程、多线程和多协程三种下载方法的性能进行比较分析, 为相关 GNSS 数据下载软件开发提供参考和借鉴。

## 1 GNSS 数据下载方法

GNSS 数据主要存储于目标 IGS 数据中心和 CORS 对应的服务器内, 用户可通过网络请求的方式获取。在数据下载过程中采用单线程、多线程和多协程等方法, 可以实现数据的有效下载。其中, 线程是操作系统能够进行运算调度的基本单位, 是程序实际执行单元; 协程则是一种用户态的轻量级线程, 它是比线程占用更少资源的执行单元<sup>[12-14]</sup>。下面依次对单线程、多线程和多协程下载方法做简要介绍。

### 1.1 单线程下载

单线程下载是基于单个线程的控制流逐个下载目标数据。主要流程是先向服务器发送下载请求, 建立计算机与服务器间的数据传输通道; 然后 CPU 将 web 端数据缓存到电脑主存储器 (内存) 中, 存储完毕后 CPU 被释放; 最后, 存储器直接存取 (DMA) 负责把内存里的缓存数据写入磁盘。详细步骤如图 1 所示。

收稿日期: 2021-12-06

资助项目: 国家自然科学基金项目 (41804020); 东华理工大学资源与环境经济研究中心 (21JDGL03); 广西空间信息与测绘重点实验室研究基金资助 (19-185-10-10); 东华理工大学研究生创新基金 (YC2021-S631); 东华理工大学博士科研启动基金 (DHBK2019182)

通信作者: 卢立果 E-mail: lglu66@163.com

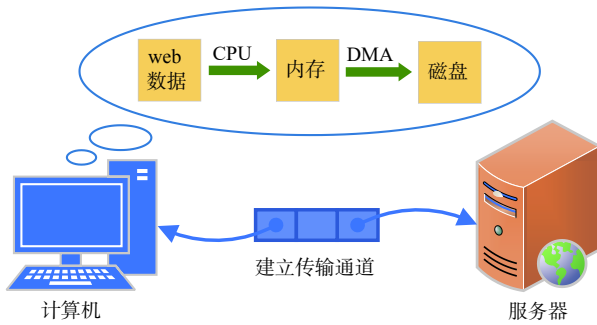


图 1 单线程下载流程图

1.2 多线程下载

多线程下载是同时开启多个线程, 每个线程用于执行不同的下载任务. 多线程下载首先以计算机和服务器间的数据传输通道为基础, 借助计算机内核的调度交替获得 CPU 的处理资源, 将每个线程任务对应的 web 数据分别缓存至内存, 最后由 DMA 将其写入磁盘, 完成多任务并发下载<sup>[15]</sup>, 如图 2 所示.

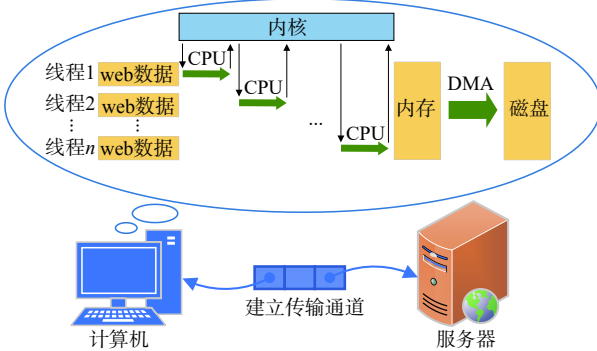


图 2 多线程下载流程图

1.3 多协程下载

多协程下载是同时创建多个协程, 每个协程负责一个下载任务. 它是以计算机和服务器间的数据传输通道为基础, 将单个协程和线程进行对接, 完成 CPU 缓存 web 数据的作业步骤; 当协程遇到 I/O(硬盘/内存) 读写操作时, 协程和线程会分离, 其中协程独自等待 DMA 将缓存数据写入磁盘, 线程则准备与下一个协程进行对接, 循环上述流程, 如图 3 所示.

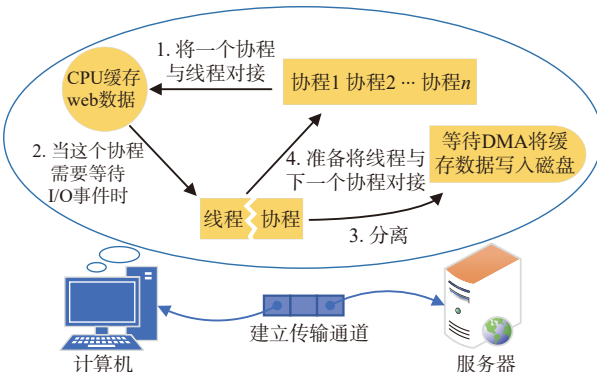


图 3 多协程下载流程图

2 数据下载与分析

为比较分析单线程、多线程和多协程三种下载方法的性能, 选取武汉大学 IGS 数据中心 2020 年 001 天至 2021 年 234 天 (共计 600 天) 的数据, 统计三种方法分别下载 100、200、...、600 天不同量级数据的耗时情况 (取 5 次测试结果的平均值). 其中, 下载测试的文件根据 GNSS 数据整体大小分为小、中、大三个量级, 小文件是指单个文件大小小于 1 MB/天的 GNSS 数据, 主要有广播星历、精密星历、地球自转参数、电离层延迟和对流层延迟等, 测试选取的是最常用的 GPS 广播星历文件; 中文件是指单个文件大小介于 1~4 MB/天的数据, 该类文件包括大部分的观测数据, 测试选取 BJFS 站 RINEX 2.xx 观测数据; 大文件是单个文件大小超过 4 MB/天的数据, 它在 GNSS 数据中占极少部分, 主要有天线相位中心改正和部分观测文件等, 测试选取 WUH2 站 RINEX 3.xx 观测数据. 不同时间跨度下的小、中、大三类文件总大小如表 1 所示, 三种下载方法的耗时结果如表 2 所示.

表 1 不同时间跨度的文件大小统计 MB

文件数量	100天	200天	300天	400天	500天	600天
小	6.4	12.7	18.9	24.7	30.1	35.5
中	163.2	329.1	492.6	648.4	801.5	943.3
大	466.3	1 024.9	1 587.4	2 099.8	2 539.3	3 102.7

表 2 不同时间跨度的三种下载方法耗时统计 s

方法名称	文件量级	100天	200天	300天	400天	500天	600天
单线程	小	68	135	211	269	354	416
	中	122	245	365	479	597	711
	大	178	392	604	795	964	1 178
多线程	小	5	8	11	15	19	23
	中	22	43	62	80	97	114
	大	67	146	223	292	346	410
多协程	小	16	30	44	56	68	79
	中	29	54	78	100	122	142
	大	73	144	212	274	326	396

文中三种方法下载耗时采用课题组研发的

“GNSS 数据下载软件 (GDDS v1.0)”进行统计, 该软件采用 Python 设计开发, 支持 IGS 和区域 CORS 数据的下载 [16]。测试所用 PC 机硬件配置为: Intel Core i5-7200 CPU, 2.50 GHz 主频, 8 GB 内存, Win10 操作系统 (64 位); 平均网速为: 12.5 MB/s。

### 2.1 单线程下载结果分析

图 4 为单线程下载不同量级文件。由图 4 可知, 单线程在下载单个文件量级相同的任意时间跨度的数据时, 速度几乎保持不变; 随着下载任务中单个文件量级逐渐增大, 单线程的下载速度也有提升。

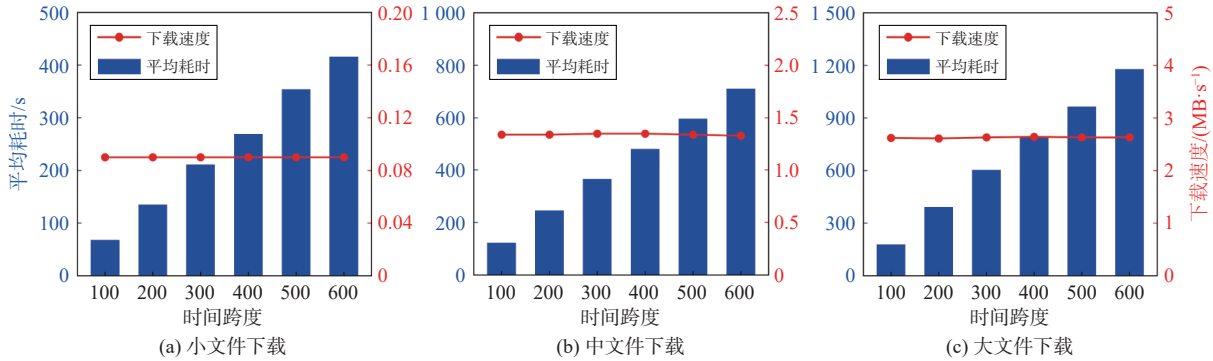


图 4 单线程下载不同量级文件

文件下载的耗时主要来源于服务器响应、CPU 缓存和 DMA 拷贝三个等待过程, 其中服务器响应由其自身硬件配置决定, 与请求下载的文件大小无关; CPU 缓存和 DMA 拷贝受文件大小和相关操作速度的双重影响。在单线程下载过程中, 一直循环执行着等待服务器响应下载请求、CPU 缓存 web 数据和 DMA 将数据拷贝至磁盘等耗时操作。当下载任务中的各文件大小基本一致时, 上述三个等待过程的耗时大致相同, 因此单线程下载任意时间跨度的相同量级文件的速度保持不变; 随着下载任务中单个文件量级

增大, CPU 缓存和 DMA 拷贝的耗时均线性增长, 但等待服务器响应的耗时维持不变, 因此总耗时的增长率低于文件量级的增长, 即表现出单线程下载大文件的速度优于中小文件。

### 2.2 多线程下载结果分析

图 5 是多线程下载不同量级文件的详情。由图 5 可知, 多线程在下载小文件时, 随着时间跨度的增大, 下载速度先急剧增大, 300 天小文件对应的下载速度达到峰值, 然后下载速度有一定的回落; 在下载中、大文件时, 随着时间跨度的增大, 下载速度逐渐加快。

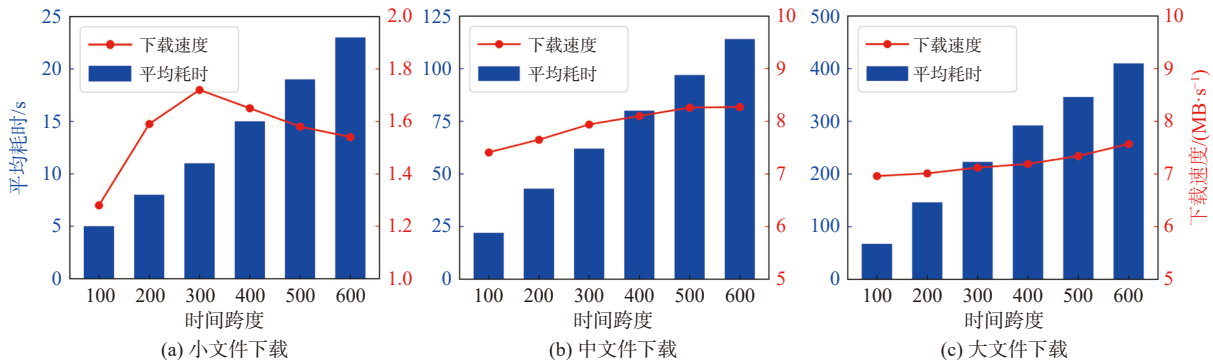


图 5 多线程下载不同量级文件

多线程是通过快速切换 CPU 资源达到 CPU 缓存和 DMA 拷贝同时进行的效果, 因此在下载小文件过程中, 多线程能短时间完成大批量数据下载工作, 显著提高整体下载速度; 当下载速度达到服务器允许此类型文件的最快传输值时, 出于安全考虑服务器会主动限制该 IP 地址的后续数据传输速度, 导致长时间跨度的下载任务速度降低。在下载中、大文件时, 多线程下载速度难于达到服务器此类型数据的报警

值, 所以下载速度会随时间跨度的增大而加快。

### 2.3 多协程下载结果分析

图 6 是多协程下载不同量级文件的详情。由图 6 可知, 多协程的下载速度先随时间跨度的增大而加快, 当时间跨度达到一定程度后, 下载速度便稳定下来。多协程的这种速度变化趋势随着下载任务中单个文件量级的增大愈加显著。



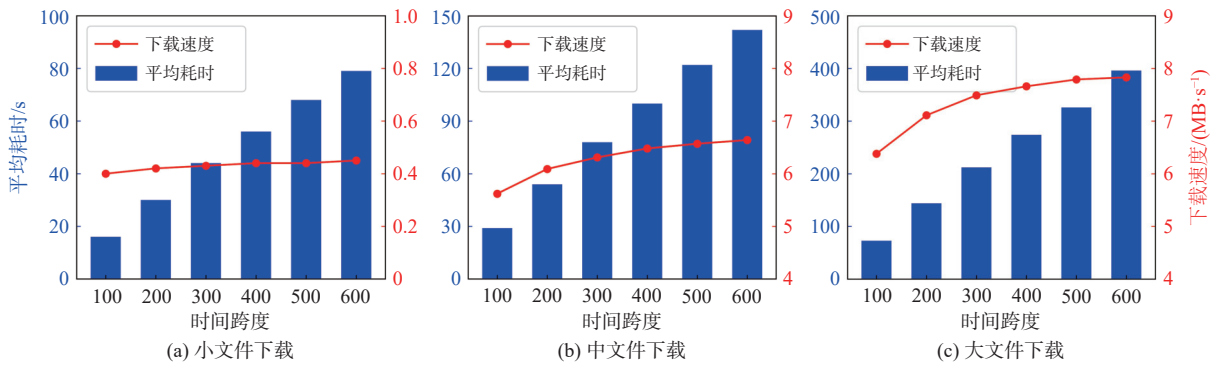


图 6 多线程下载不同量级文件

多线程下载是充分利用阻塞的时间实现多任务的分段式下载,即在执行下载任务中,多线程遇到 I/O 等待便会挂起当前任务,切换至下一个任务执行;当 I/O 处理完毕后,又切换回该任务继续运行。多线程在下载过程中充分利用了 CPU 资源,因此其下载的文件数量越多,理论下载速度就越快。但在实际下载过程中,多线程的下载速度并不会随下载的文件数量增多而无限加快,它会受到一些其他因素的制

约,如信道带宽、计算机性能等,导致下载速度到达一定程度后增长缓慢。

## 2.4 三种方法综合分析

图 7 是三种方法在小、中、大三类 GNSS 数据下载中的速度对比。由图 7 可知,在下载中小文件时,下载速度最快的是多线程,其次是多线程,最后是单线程;下载大文件时,多线程和多线程的下载效果相近,均优于单线程下载,其中多线程的速度略高于多线程。

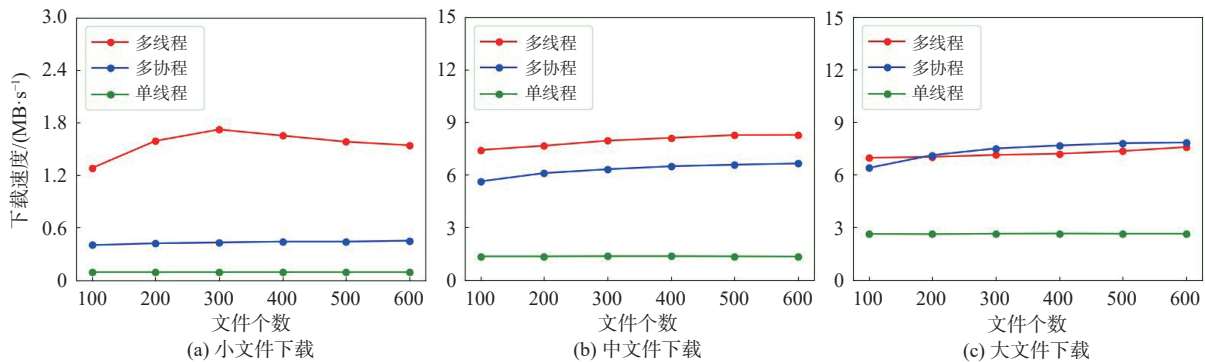


图 7 三种方法下载速度对比

单线程下载是基于下载步骤有序运行,整个下载过程 CPU 大部分时间都处于空闲状态;多线程是通过内核调度完成 CPU 的支配,让 CPU 保持连续运行的状态;多线程是根据用户需求让 CPU 处理不同的任务片段,使其一直处于工作状态。由此可知单线程的 CPU 利用率明显低于多线程和多线程,需花费更多时间完成相同的 web 数据缓存工作,导致单线程下载不同量级文件的速度都慢于多线程和多线程。

多线程利用的是 CPU 多个核参与下载,缺点在于线程的切换是基于内核的调度完成,所以其切换的开销较大。多线程在下载过程中利用的是单核 CPU,优点在于线程间的切换是寄存器上下文和栈的恢复,不占用内核的开销,故线程间的切换速度快于线程。在下载中小文件时,多线程借助调用 CPU 多核参与下载的特性,其下载速度明显高于多线程;随着下载任务中单个文件量级的增大, CPU 多核的性能优势

在下载过程中逐渐弱化,出现多线程与多线程下载速度相互靠拢的现象;在下载大文件时,两者都因外部条件限制达到最佳速度,而线程间的切换快于线程,所以多线程下载速度整体略优于多线程。

## 3 结语

本文对单线程、多线程和多线程三种 GNSS 数据下载方法进行了对比分析。结果表明:下载中小文件时多线程明显优于其他两种方法;下载大文件时多线程略优于多线程,且两者的性能都高于单线程;由于 GNSS 数据主要以中小型文件为主,所以 GNSS 数据下载软件宜采用多线程下载方式。

## 参考文献

- [1] 杨元喜,杨诚,任夏. PNT智能服务[J]. 测绘学报, 2021, 50(8): 1006-1012.

- [2] TEUNISSEN P J G, MONTENBRUCK O. Springer handbook of Global Navigation Satellite Systems [M]. Springer International Publishing, 2017. DOI:10.1007/978-3-319-42928-1-17
- [3] HU Z Y, FAN L, WANG C, et al. More reliable global ionospheric maps combined from ionospheric products of the seven IGS analysis centers[J]. *Results in physics*, 2020, 17(3-4): 103162. DOI: 10.1016/j.rinp.2020.103162
- [4] 杨飞, 郭际明, 李奔韬, 等. GPT3模型估计对流层产品精度检验与分析[J]. *大地测量与地球动力学*, 2021, 41(11): 1123-1126.
- [5] 马一方, 周晓慧, 华培学. 测站坐标阶跃对参考框架稳定性的影响分析[J]. *测绘科学*, 2021, 46(10): 20-24.
- [6] 魏二虎, 刘学习, 孙浪浪, 等. 测站数目和观测弧段对GPS解算地球自转参数的影响分析[J]. *大地测量与地球动力学*, 2017, 37(2): 187-191.
- [7] 马俊, 姜卫平, 周晓慧, 等. 联合小波和方差分量估计方法分析中国IGS测站时间序列变化特征[J]. *武汉大学学报(信息科学版)*, 2018, 43(4): 629-636.
- [8] 倪蓉蓉, 王庆, 潘树国, 等. IGS服务数据专用FTP客户端设计与实现[J]. *测绘科学*, 2011, 36(6): 278-280,288.
- [9] 王丞, 李磊, 张铭彬. 一种高效的GNSS数据自动化下载方法的实现[J]. *全球定位系统*, 2018, 43(3): 129-134.
- [10] CHEN Z S, CUI Y, LI L Y, et al. GDP: an open-source GNSS data preprocessing toolkit[J]. *GPS solutions*, 2020, 24(3): 42-49. DOI: 10.1007/s10291-020-01003-z
- [11] 王建伟, 郭春喜, 白穆, 等. 海量IGS数据实时线程池并发获取[J]. *测绘通报*, 2020(12): 93-96,105.
- [12] 周维, 周可人, 栾钟治, 等. 基于共享内存的多核时代数据结构研究[J]. *软件学报*, 2016, 27(4): 1009-1025.
- [13] 蒋炎岩, 许畅, 马晓星, 等. 获取访问依赖: 并发程序动态分析基础技术综述[J]. *软件学报*, 2017, 28(4): 747-763.
- [14] 刘书健. 基于协程的高并发的分析与研究[D]. 昆明: 昆明理工大学, 2016.
- [15] 李鹏, 曾露, 王焕东. 多核处理器系统I/O访存优化研究[J]. *高技术通讯*, 2018, 28(6): 488-499.
- [16] 吴汤婷, 梁乔, 胡伟建, 等. GNSS数据下载软件设计与实现[C]//第十三届中国卫星导航年会, 北京, 2022.

### 作者简介

吴汤婷 (1990—), 女, 博士, 讲师, 主要研究方向为 GNSS 数据处理.

梁乔 (1998—), 男, 主要研究方向为 GNSS 数据处理.

陈本富 (1968—), 男, 硕士, 主要研究方向为测量数据处理.

卢立果 (1989—), 男, 博士, 讲师, 主要研究方向为 GNSS 数据处理.

## Comparative analysis of three GNSS data download methods

WU Tangting<sup>1,2</sup>, LIANG Qiao<sup>2</sup>, CHEN Benfu<sup>2</sup>, LU Ligu<sup>3</sup>

(1. *Research Center of Resource and Environment Economics, East China University of Technology, Nanchang 330013, China;*

2. *Faculty of Geomatics, East China University of Technology, Nanchang 330013, China;*

3. *Guangxi Key Laboratory of Spatial Information and Geomatics, Guilin University of Technology, Guilin 541006, China)*

**Abstract:** In order to comprehensively evaluate the performance of single-thread, multi-thread, and multi-coroutine in Global Navigation Satellite System (GNSS) data download, this paper uses the International GNSS Service (IGS) data center of Wuhan University as the download source and analyzes the download effects of three methods in terms of the time span and magnitude of the download file. The results show that in the process of small and medium-sized files download, multi-thread has more advantages than single-thread and multi-coroutine, while for large file downloads, the download effect of multi-coroutine is slightly better than multi-thread, and both have better performance than that of single-thread. Since GNSS data file is generally small and medium-sized file, multi-thread is more suitable for GNSS data download.

**Keywords:** Global Navigation Satellite System (GNSS); data download; single-thread; multi-thread; multi-coroutine